Paper:

Intelligent Automated Guided Vehicle Controller with Reverse Strategy

Shigeru Kato* **and Kok Wai Wong****

*Niihama National College of Technology
7-1 Yagumo-cho, Niihama-shi, Ehime 792-8580, Japan
E-mail: skatou@ele.niihama-nct.ac.jp
**Murdoch University
South St., Murdoch, WA 6150, Australia

[Received December 22, 2010; accepted March 12, 2011]

This paper describes the intelligent Automated Guided Vehicle (AGV) control system using Fuzzy Rule Interpolation (FRI) method. The AGV used in this paper is a virtual vehicle simulated using computer. The purpose of the control system is to control the simulated AGV by moving along the given path towards a goal. Some obstacles can be placed on or near the path to increase the difficulties of the control system. The intelligent AGV should follow the path by avoiding these obstacles. This system consists of two fuzzy controllers. One is the original FRI controller that mainly controls the forward movement of the AGV. Another one is the proposed reverse movement controller that deals with the critical situation. When the original FRI controller faces the critical situation, our proposed reverse controller will control the AGV to reverse and move forward towards the goal. Our proposed reverse controller utilizes the advantage of FRI method. In our system, we also develop a novel switching system to switch from original to the developed reverse controller.

Keywords: fuzzy rule interpolation (FRI), automated guided vehicle (AGV), obstacle avoidance

1. Introduction

Fuzzy logic is an intelligent method that can handle complex problem with uncertainties [1]. In some control systems, the controller sometimes faces with unexpected situations due to the complexity of the problem. Under some unexpected situations, the modification of the controller is necessary to handle those situations. However adding new fuzzy rules may increase the complexity of the rule base and thus increase the computational time.

In order to address the issue of the complexity, there is an attempt to use techniques that use small number of significant rules. One of them is the use of sparse rule base. In ordinal fuzzy controller [2], if the rule base is sparse (not complete), there could be an observation, which does not find any fuzzy rule to fire. In this case, Fuzzy Rule Interpolation (FRI) is necessary to be used [3–5]. When using FRI, it is not necessary to cover the entire universe of discourse of the antecedent parts of the fuzzy rules. FRI can provide reasonable conclusions even though none of the existing fuzzy rule fires for the current observation. However, depending on the neighbourhood fuzzy memberships, FRI method may face unexpected situation in which large rule modification may be required. In view of this problem, other control module that can complement FRI can be used to solve the new problem. In this paper, we presented a hybrid system which consists of FRI controller and the classical fuzzy controller used to deal with the unexpected situations.

To illustrate the method proposed in this paper, the controller designed is used to control an Automated Guided Vehicle (AGV) [6]. The AGV used in this paper is a virtual vehicle, for example a car that is simulated using computer. The purpose of the control system is to control the simulated AGV moving along a given guided path towards the goal known as docking station. Some obstacles can be placed on or near the path to increase the difficulties of the control system.

The intelligent AGV should trace the guided path by avoiding these obstacles. In order to control the AGV, fuzzy reasoning and automata with FRI technique have been used [7]. Although the original fuzzy controller is sufficient to control AGV in most foreseeable situations, sometime the AGV will collide with the obstacles and get stuck in some unexpected situations. To handle such unexpected situations, it is essential to introduce the reverse strategy to the control system.

For comparison purposes, we used three different intelligent AGV controllers. For the first controller (i.e., Controller A), it is the original implementation of the AGV using FRI (Fuzzy Rule Interpolation) method [7]. However in the original AGV control system, they only use forward controlled strategy to avoid the obstacles. In the second controller (i.e., Controller B), we connect the reverse strategy module to Controller A. However Controller B is not sufficient to arrive at the docking station (goal). In the third controller (i.e., Controller C), we call the hybrid intelligent controller in which the reverse strategy module can adjust the original FRI rules to arrive at the docking station.

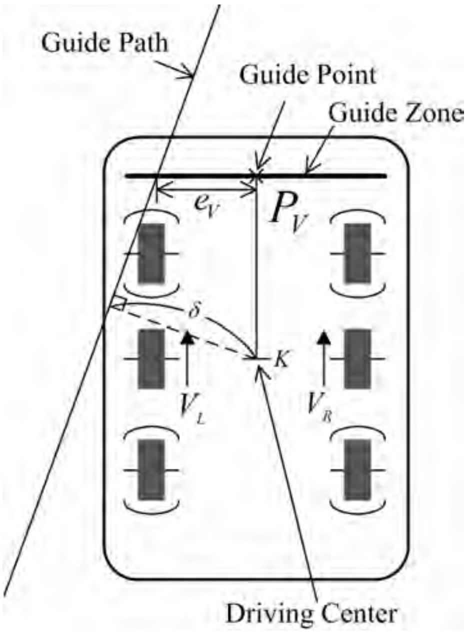

Fig. 1. Structure of AGV.

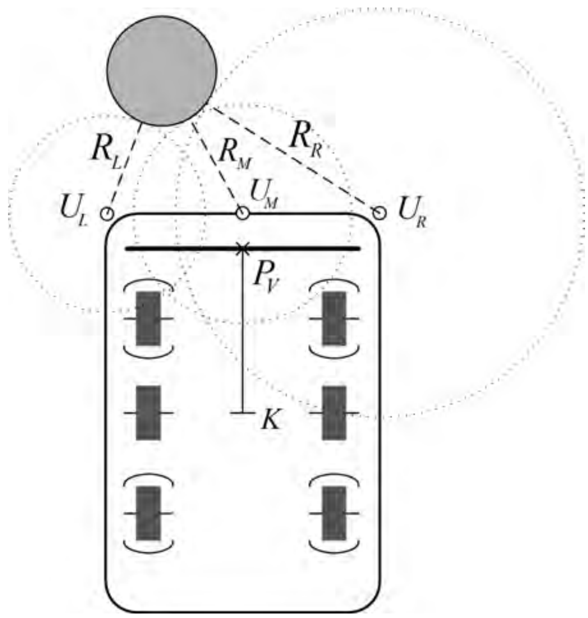

Fig. 2. Obstacle sensor.

The paper is organized as follows. Section 2 takes a look at the automated guided vehicle. Section 3 gives the concepts behind the obstacles avoidance of the simulated AGV. In Section 4, the details of original implementation of obstacle avoidance by forward movement are discussed. Section 5 describes reverse strategy to solve critical situation. Section 6 presents the test results of the propose obstacle avoidance. Finally, Section 7 provides some discussions and conclusions.

2. Automated Guided Vehicle

2.1. Overview of Automated Guided Vehicle

An AGV as shown in **Fig. 1** is an intelligent driver-less machine which is capable of determining its motion status according to the environmental condition. It is equipped with an on-board controller which enables it to process information obtained from the sensors that are attached on the vehicle. In this paper, the AGV has two fixed directional wheels and four free directional wheels as shown in **Fig. 1**. The AGV can move and turn by controlling the contour speed V_L and V_R of two fixed directional wheels. In the cases $V_L = V_R$, $V_L > V_R$ and $V_L < V_R$, the AGV goes straight, turns right and turns left, respectively [8].

2.2. Guide Path Tracking Sensor

The guide path is usually a painted marking or a passive or active wire (guide wire) glued onto or built into the floor. The main goal of the AGV is to follow the marking of the guide path. The guiding system senses the position of the guide path by special sensors (guide zone) tuned for the given guide path. The guide zone is a section of the AGV as shown in **Fig. 1**. The distance e_V between the guide path and the guide point is calculated. The distance δ between the guide path and the driving centre (called path tracking error) is calculated from the previous e_{V0} and the current value e_V and from the move of the AGV [6].

2.3. Obstacle Sensor

There are three ultrasonic distance sensors on the front of the AGV, one in the middle (U_M) and one-one on both sides (U_L, U_R) as shown in **Fig. 2**. The distances between each sensor and obstacle are given as R_M, R_L and R_R, respectively [9].

3. Information for Path Tracking and Collision Avoidance

The main goal of the steering control is path tracking (to follow a guide path). To make the example task more complex, we added a second goal as restricted (limited) collision avoidance. The restricted collision avoidance means "avoiding obstacles without risking the chance of losing the guide path."

3.1. Information for Path Tracking

The base idea of the path tracking is very simple: keep the driving centre K of the AGV as close as possible to the guide path, and then if the driving centre is close enough to the guide path, simply turn the AGV into the direction of docking station.

The above simple strategy needs only two observations: δ path tracking error, and e_V the distance between the guide path and the guide point. Using the guide zone we can determine e_V, but we have no information on the path tracking error δ. Therefore estimated momentary path tracking error δ is calculated from the previous (e_{V0}) and the current value (e_V) and from the move of the AGV [6].

3.2. Information for Collision Avoidance

In order to define the collision avoidance, we have to study the different types of possible collision situations. There are two different collision situations, the frontal and the side collision. It is sufficient to have three ultrasonic distance sensors on the front of the AGV as shown in **Fig. 2**. The three distances (R_L, R_M, R_R) are measured by the three obstacle sensors. (R_L, R_M, R_R) give sufficient information in finding a strategy to be able to avoid the frontal collision situations. However the observations to avoid the side collision are not so simple. Having the preconditions of motionless and avoidable obstacles, we have a chance to use the obstacle distance measurements of the near past for scanning the boundaries of the obstacles. Collecting the previous measurements of the left and right obstacle sensors and the corresponding positions of the AGV (measured by the motion sensors on the wheels), we can approximate the boundaries of the obstacles by discrete points. We call these points *unsafe*, or *risky points*. The distance measured by an obstacle sensor means the existence of a potential obstacle outside the circle defined by the position of the sensor and the measured value (see **Fig. 3**). Having more measurements and more positions, we can approximate the boundaries of the obstacles by the pair point of intersection of these circles (see **Fig. 3**). The main idea of the side collision avoidance strategy is to avoid unsafe points. In order to have observations easier to be handled, we calculate the actual maximal left and right turning angle without side collision $(\alpha_{ML}, \alpha_{MR})$ (see **Fig. 4**). These values are normalized to [0,1] by being divided by $\pi/2$ [9].

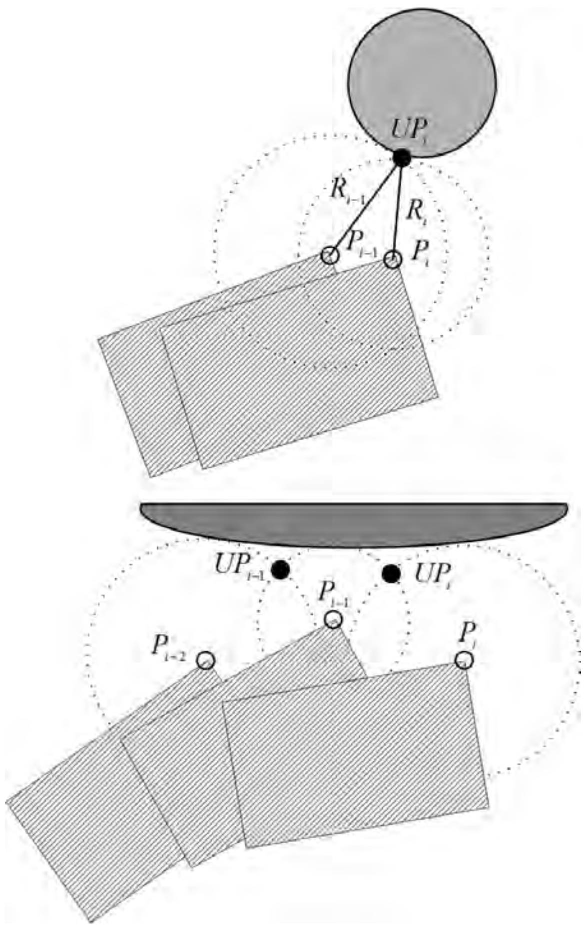

Fig. 3. Unsafe point.

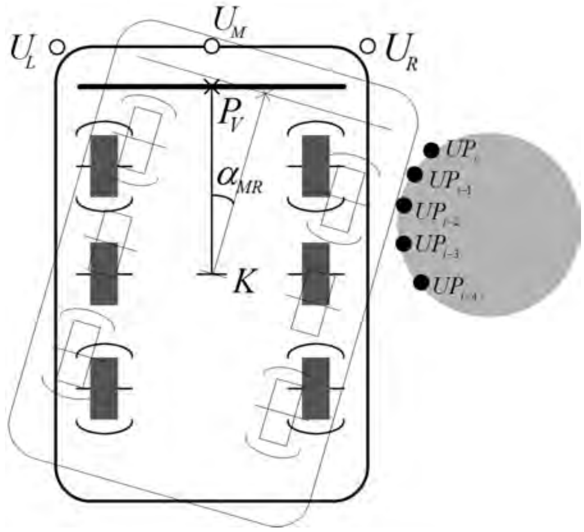

Fig. 4. Maximal right turning angle α_{MR}.

4. Forward Control Part

Figure 5 shows the system structure to control AGV. The controller consists of the forward control part (i.e., original fuzzy controller [7]) handling forward movement, and the proposed reverse controller dealing with reverse movement. These parts output V_a and V_d which are input of the AGV. V_a is the required speed of the AGV and V_d is the required steering degree. V_L and V_R are determined from the required V_a and V_d [8]. When V_a is positive value, AGV runs forward. When V_a is negative value, AGV runs backward. The bigger V_a is inputted to the AGV, the faster the AGV runs. When V_d is positive value, AGV turns right. When V_d is negative value, AGV turns left. The higher degree of V_d is given to the AGV, the bigger the angle of turn.

The input to the two parts is sensed information by AGV sensor and calculated once after it moved, such as e_V, δ, R_L, R_M, R_R, α_{ML} and α_{MR}. Only one input to the AGV is selected from the outputs of controllers depending on the situation.

Forward control part (i.e., original fuzzy controller [7]) has four Fuzzy Logic Controllers (FLCs) and each controller outputs each conclusion of (V_a, V_d) as shown in **Fig. 5**. The FLC has each partially valid strategy to control the AGV in a particular situation. These four

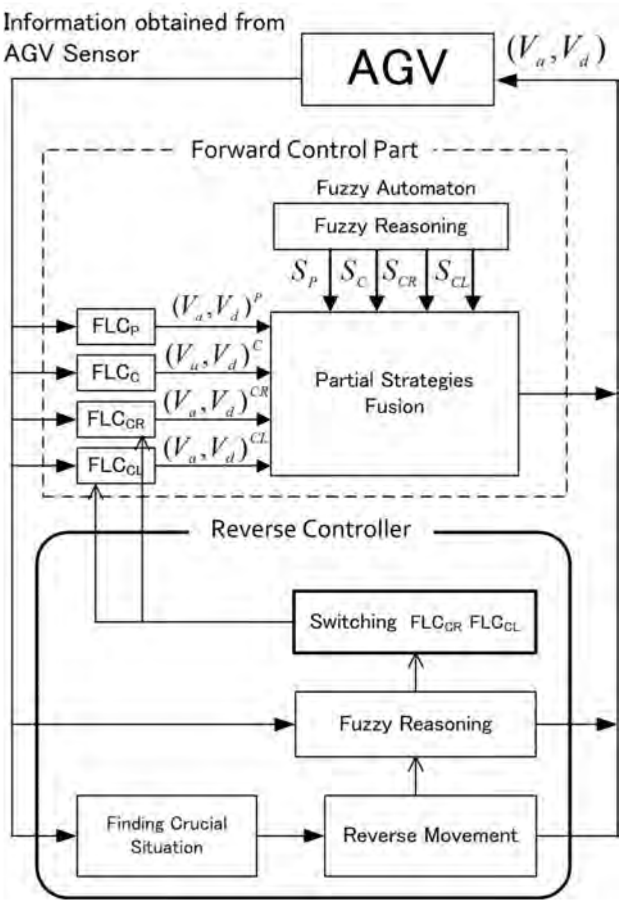

Fig. 5. Propose controller.

strategies are "*The path tracking and restricted collision avoidance strategy (FLC_P)*," "*The collision avoidance strategy (FLC_C)*," "*The collision avoidance with left tendency strategy (FLC_{CL})*" and "*The collision avoidance with right tendency strategy (FLC_{RC})*" [7].

Finally, one conclusion of (V_a, V_d) is obtained by considering the output of fuzzy automaton. S_P, S_C, S_{CR} and S_{CL} express the similarities between the AGV state and the prerequisite of each strategy, respectively [7].

4.1. The Path Tracking and Restricted Collision Avoidance Strategy

The FLC_P has *the path tracking and restricted collision avoidance strategy*. The main goal of this strategy is following guide path. And sub-goal is avoiding obstacles without risking the chance of losing the guide path. The rules for this strategy for V_d and V_a are shown in **Table 1**. It may seem that the number of rules is smaller than ordinal. As Fuzzy Rule Interpolation (FRI) technique [3–5] is used, reasonable conclusions can be obtained from small number of significant rules for this part of the control. Although the rules 1 and 2 in **Table 1** do not correspond to previous work [7], the rules in **Table 1** has been tested and the AGV can be controlled by these rules adequately.

Table 1. The path tracking and restricted collision avoidance strategy.

	e_V	δ	R_L	R_R	R_M	α_{ML}	α_{MR}	V_d
1	NL						L	PL
2	PL					L		NL
3	NM	Z					L	PL
4	PM	Z				L		NL
5	NM	PM	L		L	L		Z
6	PM	NM		L	L		L	Z
7	Z	PM	L		L	L		NS
8	Z	NM		L	L		L	PS
9	Z	PM	S		S			PL
10	Z	NM		S	S			NL
11	Z	Z	L	S	S			NL
12	Z	Z	S	L	S			PL

	e_V	δ	R_L	R_R	R_M	V_a
1	Z	Z	L	L	L	L
2	NL	PL				Z
3	PL	NL				Z
4	NL	Z				Z
5	PL	Z				Z

N: negative, P: positive, L: large, M: middle, S: small, Z: zero

Table 2. The collision avoidance strategy.

	R_L	R_R	R_M	α_{ML}	α_{MR}	V_d
1		Z		L		NL
2	Z				L	PL
3		Z	L	S		NVS
4	Z		L		S	PVS

	R_L	R_R	R_M	V_a
1	L	L	L	L
2			S	S

N: negative, P: positive, L: large, M: middle, S: small, VS: very small, Z: zero

4.2. The Collision Avoidance Strategy

FLC_C has *the collision avoidance steering strategy*. The rules for this strategy for V_d and V_a are shown in **Table 2**.

4.3. The Collision Avoidance with Left/Right Tendency Strategy

FLC_{CL} and FLC_{CR} have *the collision avoidance with left tendency strategy* and *the collision avoidance with right tendency strategy*, respectively. These two partially valid strategies are basically the same as *the collision avoidance strategy*, except left or right turning tendencies in case of no left or right turning difficulties. These strategies are needed to aid the finding of the path after leaving it (because of the obstacles). Their rule bases are the same as the rule bases of *the collision avoidance strategies*, except this time it has two additional rules. The additional rules for the left and right tendency to perform the collision avoidance steering strategy for V_d are shown in **Tables 3** and **4**, respectively.

Table 3. The collision avoidance with left tendency strategy.

	R_L	R_R	R_M	α_{ML}	α_{MR}	V_d
1-4	…	…	…	…	…	…
5				S		Z
6	L		L	L		NL

Table 4. The collision avoidance with right tendency strategy.

	R_L	R_R	R_M	α_{ML}	α_{MR}	V_d
1-4	…	…	…	…	…	…
5					S	Z
6		L	L		L	PL

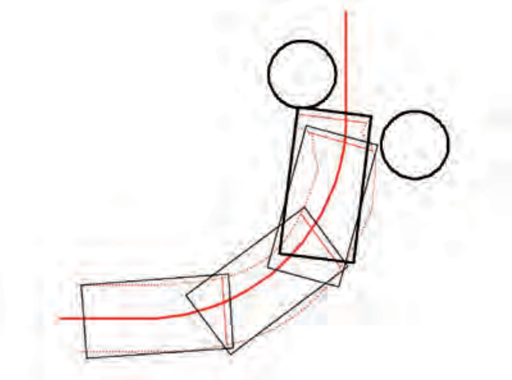

Fig. 6. The result of collision without reverse.

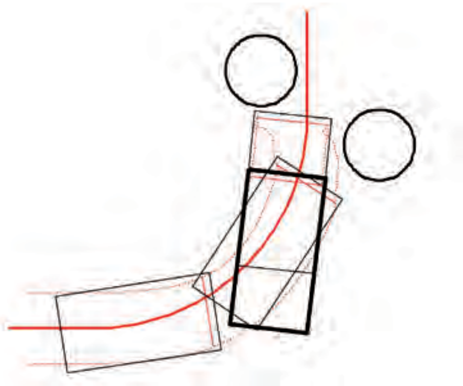

Fig. 7. The result of reverse.

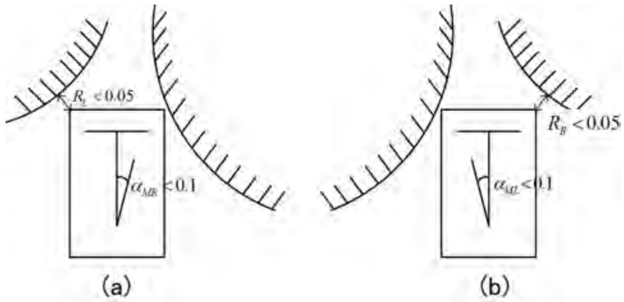

Fig. 8. The crucial situation.

5. Reverse Controller

This controller deals with particular situation when forward control part encountered some unexpected situations. For example **Fig. 6** shows the control result by using only the forward control part, i.e., the original AGV controller [7]. In this case, the forward control part cannot handle this situation appropriately (accordingly the AGV collides with the obstacle) without the modification of the existing fuzzy rules. To avoid such collision, we know that the AGV should simply go backward (reverse) before collision. However the reverse behavior is not taken into account when designing the original fuzzy control. The rule base only considers forward movement. In order to take such behavior into account, we propose by simply adding another module for controlling AGV to reverse [10–12]. **Fig. 7** shows the example of the reverse behavior where dotted line is trajectory of AGV and the bold lined square shows the AGV while reversing.

When the AGV falls into crucial situation where the AGV cannot go forward any more, the reverse controller commands the AGV to reverse backward from the path. The crucial situation is defined as $R_L < 0.05 \wedge \alpha_{MR} < 0.1$ or $R_R < 0.05 \wedge \alpha_{ML} < 0.1$ as shown in **Fig. 8**, respectively.

For example, in **Fig. 8(a)** the AGV cannot turn left because R_L is very small. On the other hand α_{MR} (the maximal right angle without collision) is very small therefore the AGV cannot turn right. Finally the AGV has no choice except to reverse and then stir forward to the other direction.

When the AGV finds the crucial situation, the AGV reverses until R_L, R_R and R_M become higher than 0.6, which is the safe situation without collision risk.

After reversing, the AGV should head towards to the docking station (goal). Therefore this part needs to have the essential rule to determine next behavior of the AGV. After various reversing situations are considered, we then applied fuzzy inference in which reasonable conclusion would be obtained from the linguistic rules [2].

5.1. Forward Movement Path After Reverse

After reversing, the AGV can choose the two paths. One is the left path along the obstacles, and another is the right path. As shown in **Fig. 9**, when $R_L < R_R$, the AGV would be better to choose the left path because the AGV is away from guide path for shorter period than by choosing right path.

Therefore when $R_L \leq R_R$, the controller performs the *left path movement*, otherwise the *right path movement*.

When the *left path movement* is selected, the following simple fuzzy rules are used to determine V_d.

IF R_L is M then V_d is NL.
IF R_L is L then V_d is NM.

When the *right path movement* is selected, the following

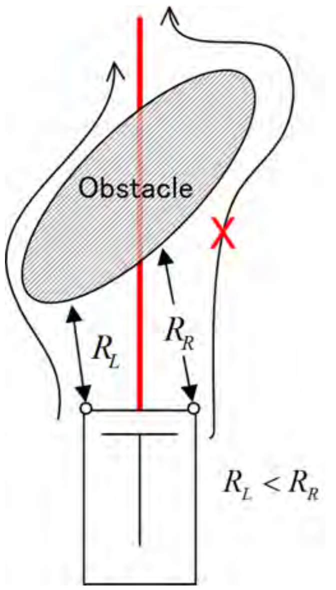

Fig. 9. Forward movement path after reverse.

Table 5. The left and right path movement fuzzy rule.

	R_L		R_R		R	
	M	L	M	L	M	L
V_d	NL	NM	PL	PM		
V_a					M	L

fuzzy rules are used to determine V_d.

IF R_R is M then V_d is PL.
IF R_R is L then V_d is PM.

V_a is determined by the following fuzzy rules regardless of the path direction.

IF R is M then V_a is M.
IF R is L then V_a is L.

R_L or R_R are substituted for R in the rule depending on the selected path movement. V_a and V_d are derived by Mamdani calculation [2]. The rules are shown in **Table 5**.

The fuzzy sets are shown in **Fig. 10**. When V_a and V_d is determined by fuzzy inference, these values are outputted to the AGV for 1.5 sec to control the AGV going forward as avoiding obstacles. The control time is determined by tuning through some trials.

5.2. Adjustment of FRI Rule Basis

Although the AGV can get out of critical situation, it may not necessarily arrive at the docking point. **Fig. 11** shows the control result where the AGV collides with the obstacle. The AGV is controlled by the original FRI system in [7] without reverse strategy.

Figure 12 shows the example by original FRI system with the reverse strategy in the same condition as in **Fig. 11**. However the left side of AGV collides with the obstacle after going beyond the obstacles. This is because

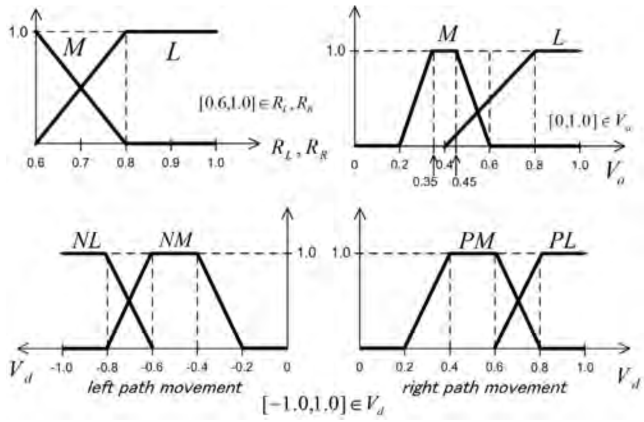

Fig. 10. Fuzzy sets for left and right path movement.

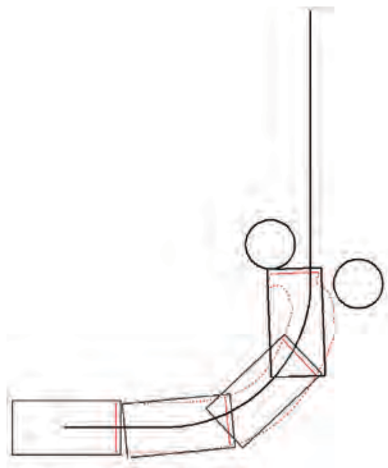

Fig. 11. Frontal collision by original FRI controller.

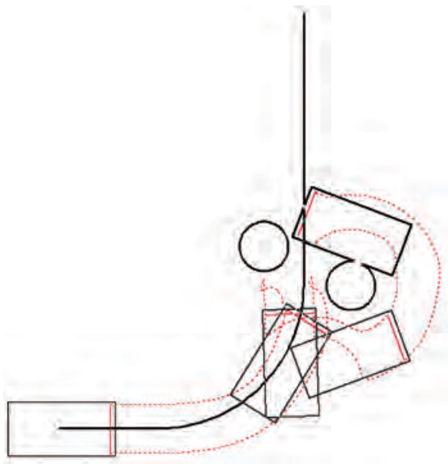

Fig. 12. Side collision by reverse strategy.

the AGV turns left tightly along the left obstacle, consequently the AGV approaches the left obstacle and then the AGV collides by *the collision avoidance with left tendency strategy* (i.e., FLC_{CL}). In the same situation, the AGV sometimes not only collides with the obstacle but also loses the guide path. Such problem would also oc-

Table 6. The collision avoidance with left tendency strategy after reverse.

	R_L	R_R	R_M	α_{ML}	α_{MR}	V_d
1-5	...	...	...	...	...	...
6	L		L	L		NM

Table 7. The collision avoidance with right tendency strategy after reverse.

	R_L	R_R	R_M	α_{ML}	α_{MR}	V_d
1-5	...	...	...	...	...	...
6		L	L		L	PM

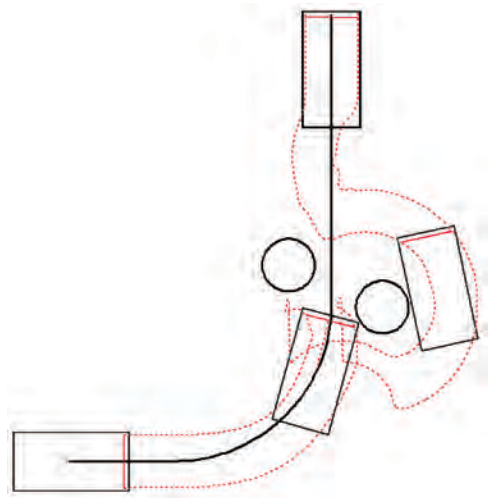

Fig. 13. Reverse strategy by adjusting original FRI.

cur by *the collision avoidance with right tendency strategy* (i.e., FLC_{CR}). Therefore after moving forward, the FLC_{CL} and FLC_{CR} in the original FRI system should be switched over to the reverse FLC (see **Fig. 5**) as shown in **Tables 6** and **7**.

Only the consequent parts of the 6th rule are switched from NL (Negative Large) to NM (Negative Middle) and from PL (Positive Large) to PM (Positive Middle), respectively.

Such modification is not easy in ordinal fuzzy controller with complicated fuzzy rule base [2], because we have to take care of covering the entire input universe of discourse by the fuzzy sets, i.e., have to know all the details of the controller. The FLC constructed by FRI (Fuzzy Rule Interpolation) method deals with the complex problem by using small number of significant rules. In this technique it is not necessary to cover the entire input universe of discourse [3–5]. Based on this principle, sparse fuzzy rule base can be created. Therefore we can modify the significant FLCs depending on the situations without considering the full fuzzy rule base construction. The original forward movement AGV controller using FRI method is comprehensible and is easy to add new function modules.

Figure 13 shows the simulation result with the propose controller in the same condition as in **Fig. 12**. It is found that the AGV turns left loosely therefore it can arrive at the docking point.

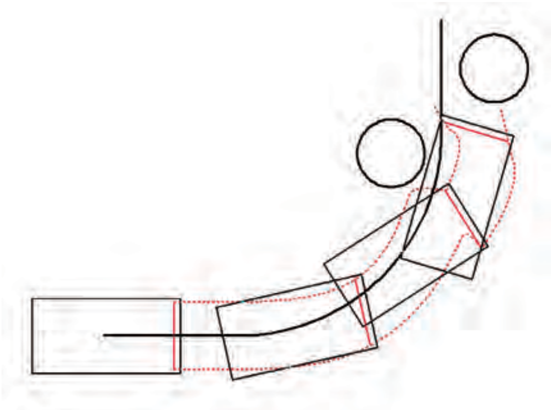

Fig. 14. Result by Controller A.

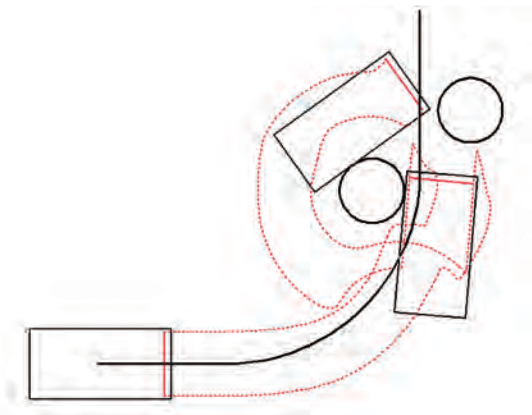

Fig. 15. Result by Controller B.

6. Simulation Experiments

In order to show the usefulness of the propose controller, simulation results are generated.

6.1. Simulation Experiment 1

The simulation experiment is performed in the condition where there are one guide path having one curve of 3.0 m radius and two obstacles around the path. The maximum speed, width and length of AGV are 1.5 m/s, 1.1 m and 2.2 m, respectively.

Figure 14 shows the result of AGV control without reverse controller (i.e., Controller A). The dotted line means the trajectory of AGV. The end of the trajectory is the point where AGV collides with obstacle. **Fig. 15** shows the result by the proposed system without the reverse FLC switching (i.e., Controller B). The condition is the same as the one in **Fig. 14**. It is found that the AGV reverses just

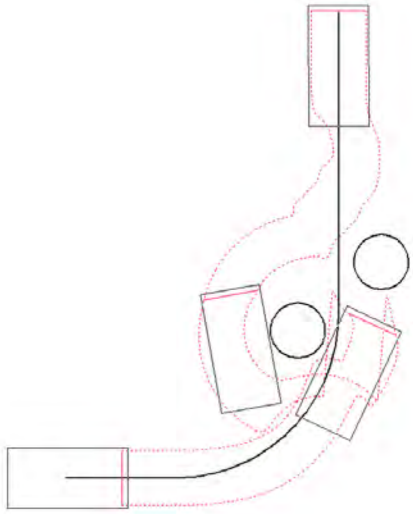

Fig. 16. Result by Controller C.

before colliding and the AGV turns left after reversing. However the right side of the AGV collides to the obstacle and does not arrive at the goal. The **Fig. 16** shows the result of the proposed system with the reverse FLC switching (i.e., Controller C). It is found that the AGV turns right loosely after reverse and arrives at the docking point.

6.2. Simulation Experiment 2

By comparing the control result of Controller A, B and C, Controller C performed the best. The condition is as shown in **Fig. 17**. There are two circlet obstacles around the path. The centers of the left and the right obstacles are placed within the each bold lined rectangle randomly. The Controller A, B and C are tested against 300 conditions. The result is shown in **Table 8**. The percentage is calculated by dividing the number of arrival times with the total number of trials, which is 300. It is found that the success rate of Controller C is 75 %, which is obviously higher than Controller A (52%). On the other hand, the difference between the arrival percentage of Controller A and B is not big. This result shows the modification of original FRI rule base could be essential.

The number of reverse is 106 out of 300 trials by Controllers B and C. After reverse, the success rate of arriving docking station is 77% by Controller C, and 29% by Controller B. It is found that the switching of FLC is very important to make sure that the AGV can lead to a goal after reversing.

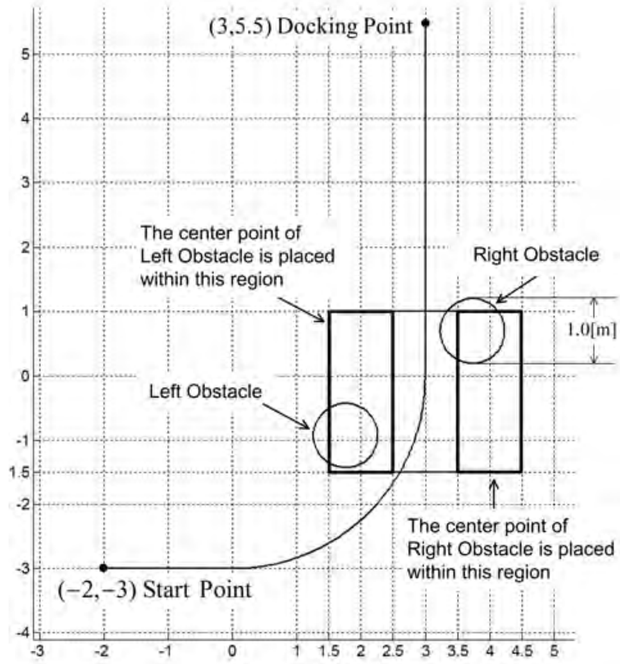

Fig. 17. Experimental condition.

Table 8. Result of simulation Experiment 2.

Controller A		Controller B		Controller C	
Fail	Arrive	Fail	Arrive	Fail	Arrive
145	155 (52%)	126	174 (58%)	75	225 (75%)
Reverse times		75	31 (29%)	24	82 (77%)
		106			

7. Discussion

This paper proposes a method to implement the reverse behavior of an intelligent vehicle. To construct intelligent obstacle avoidance, we employed existing Fuzzy Rule Interpolation (FRI) controller, which has proven to be flexible, to enhance the FLC. However, it only considers forward movement. In order not to complicate the design, we implemented a complimentary processing module (another FLC) outside the original controller to handle the reverse behavior. AGV can go beyond the obstacles by reverse behavior [10–12]. However in many situations AGV could not arrive at the docking point. In this paper, we presented an enhanced algorithm to improve the obstacle avoidance tasks. We implemented the intelligent behavior by switching original FRI controller to a reverse FLC. Since FLC (Fuzzy Logic Controller) is constructed by FRI (Fuzzy Rule Interpolation) method dealing with the complex problem by using small number of significant rules, we can design the additional function module (reverse controller) so that the significant FLCs can be adapted to the situation.

The reverse behavior of the car-like vehicle is natural and intelligent. If the autonomous agent can execute such an intelligent behavior, the variety of the movement is broadened and the agent will be authentic and look smarter. However the former approaches [13–15] in the obstacle avoidance do not focus on the realization of such natural behavior. Therefore the reverse behavior is suitable to be implemented for most autonomous agents in computer games and entertainment robots.

Our controller can guide the AGV to the goal at a higher accuracy when compared to earlier controllers by using reverse behavior for unexpected situations.

References:

[1] L. A. Zadeh, "Fuzzy Sets," Information and Control, Vol.8, Issue 3, pp. 338-353, 1965.

[2] E. H. Mamdani and S. Assilian, "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller," Int. J. of Human-Computer Studies, Vol.7, pp. 1-13, 1975.

[3] K. W. Wong, D. Tikk, T. D. Gedeon, and L. T. Kóczy, "Fuzzy Rule Interpolation for Multidimensional Input Spaces with Applications: A Case Study," IEEE Trans. of Fuzzy Systems, Vol.13, No.6, pp. 809-819, December 2005.

[4] Sz. Kovács, "Fuzzy Rule Interpolation in Practice," Proc. of the Joint 3rd Int. Conf. on Soft Computing and Intelligent Systems and 7th Int. Symposium on advanced Intelligent Systems (SCIS & ISIS 2006), (invited talk), p. 6, 2006.

[5] Z. C. Johanyak, D. Tikk, S. Kovács, and K. W. Wong, "Fuzzy Rule Interpolation Matlab Toolbox - FRI Toolbox," Proc. of IEEE Int. Conf. on Fuzzy Systems 2006, pp. 1427-1433, 2006.

[6] J. Cselényi, Sz. Kovács, L. Pap, and L. T. Kóczy, "New Concepts in the Fuzzy Logic Controlled Path Tracking Strategy of the Differential Steered AGVs," 5th Int. Workshop on Robotics in Alpe-Adria-Danube Region, RAAD'96, pp. 587-592, June 1996.

[7] Sz. Kovács, "Similarity Based Control Strategy Reconfiguration by Fuzzy Reasoning and Fuzzy Automata," Proc. of the IECON-2000, IEEE Int. Conf. on Industrial Electronics, Control and Instrumentation, pp. 542-547, October 2000.

[8] Sz. Kovács and L. T. Kóczy, "Application of the Approximate Fuzzy Reasoning Based on Interpolation in the Vague Environment of the Fuzzy Rulebase in the Fuzzy Logic Controlled Path Tracking Strategy of Differential Steered AGVs," Computational Intelligence – Theory and Applications, Lecture Notes in Computer Science, Vol.1226, Springer, pp. 456-467, 1997.

[9] Sz. Kovács and L. T. Kóczy, "Application of an Approximate Fuzzy Logic Controller in an AGV Steering System, Path Tracking and Collision Avoidance Strategy," Fuzzy Set Theory and Applications, Tatra Mountains Mathematical Publications, Mathematical Institute Slovak Academy of Sciences, Vol.16, pp. 456-467, Bratislava, Slovakia, 1999.

[10] S. Kato and K. W. Wong, "Intelligent Automated Guided Vehicle with Reverse Strategy: A Comparison Study," Lecture Notes in Computer Science 5506, Springer Heidelberg, pp. 638-646, August 2009.

[11] S. Kato and K. W. Wong, "Automated Guided Vehicle with Reverse Strategy," Australian J. of Intelligent Information Processing Systems, Vol.10, No.2, pp. 29-35, 2008.

[12] S. Kato and K. W. Wong, "The Automated Guided Vehicle Using Fuzzy Control and CBR Techniques," Proc. of Joint 4th Int. Conf. on Soft Computing and Intelligent Systems and 9th Int. Symposium on advanced Intelligent Systems, pp. 1788-1792, September 2008.

[13] J. Hoffmann, M. Jüngel, and M. Lötzsch, "A Vision Based System for Goal-Directed Obstacle Avoidance," 8th Int. Workshop on RoboCup 2004, LNAI Springer, Vol.3276, pp. 418-425, 2005.

[14] C. C. Wong, C. T. Cheng, K. H. Huang, and Y. T. Yang, "Design and Implementation of Humanoid Robot for Obstacle Avoidance," Proc. of FIRA Robot World Congress, 2007.

[15] G. Zong, L. Deng, and W. Wang, "A Method for Robustness Improvement of Robot Obstacle Avoidance Algorithm," Proc. of IEEE Int. Conf. on ROBOTICS and BIOMIMETICS, pp. 115-119, 2006.

Name:
Shigeru Kato

Affiliation:
Department of Electrical Engineering and Information Science, Niihama National Collage of Technology

Address:
7-1 Yagumo-cho, Niihama, Ehime 792-8580, Japan

Brief Biographical History:

2005 Received Engineering Doctor from University of Tsukuba
2005- Research Associate, Yuge National College of Maritime Technology
2009- Lecturer, Niihama National Collage of Technology

Main Works:

- S. Kato and K. W. Wong, "Automated Guided Vehicle with Reverse Strategy," Australian J. of Intelligent Information Processing Systems, Vol.10, No.2, pp. 29-35, 2008.
- S. Kato and T. Onisawa, "Generation of Interesting Story from Picture Information," J. of Advanced Computational Intelligence and Intelligent Informatics, Vol.11, No.7, pp. 759-766, 2007.

Membership in Academic Societies:

- Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT)
- Japan Society of KANSEI Engineering (JSKE)
- The Japan Society for Artificial Intelligence (JSAI)

Name:
Kok Wai Wong

Affiliation:
Murdoch University, School of Information Technology

Address:
South St., Murdoch, WA 6150, Australia

Brief Biographical History:

2007- Associate Professor, Murdoch University
2003- Assistant Professor, Nanyang Technological University

Main Works:

- P. Jeatrakul, K. W. Wong, and C. C. Fung, "Data Cleaning for Classification Using Misclassification Analysis," J. of Advanced Computational Intelligence and Intelligent Informatics, Vol.14, No.3, Apr. 2010, pp. 297-302, 2010.
- K. W. Wong, D. Tikk, T. D. Gedeon, and L. T. Koczy, "Fuzzy Rule Interpolation for Multidimensional Input Spaces with Applications," IEEE Trans. of Fuzzy Systems, Vol.13, No.6, December 2005, pp 809-819, 2005.
- K. W. Wong, T. D. Gedeon, and L. T. Koczy, "Construction of Fuzzy Signature from Data: An Example of SARs Pre-clinical Diagnosis System," Proc. of IEEE Int. Conf. on Fuzzy Systems – FUZZ-IEEE 2004, July 2004, Budapest, Hungary, paper 1353, 2004.

Membership in Academic Societies:

- The Institute of Electrical and Electronics Engineers (IEEE) Computational Intelligence Society
- IEEE Computer Society
- IEEE System, Man and Cybernetics
- Association for Computing Machinery (ACM)
- Australian Computer Society (ACS)
- Asia Pacific Neural Network Assembly (APNNA)